

使用&安裝說明

本作品使用godot開發，使用godot前請先安裝 .NET 6.0 (<https://dotnet.microsoft.com/en-us/download/dotnet/6.0>)。

ChatGPT在自主型代理人設計中的應用：...	2025/1/20 下午 03:45	檔案資料夾	
Godot_v4.2.1-stable_win64.exe	2025/1/20 下午 03:01	檔案資料夾	
系統使用&安裝說明.pdf	2025/1/20 下午 08:57	Microsoft Edge P...	88 KB
系統簡介.mp4	2024/11/11 上午 02:47	MP4 檔案	48,940 KB
總客.pptx	2025/1/20 下午 03:14	Microsoft Power...	12,931 KB
總客文件.pdf	2025/1/20 下午 08:52	Microsoft Edge P...	926 KB

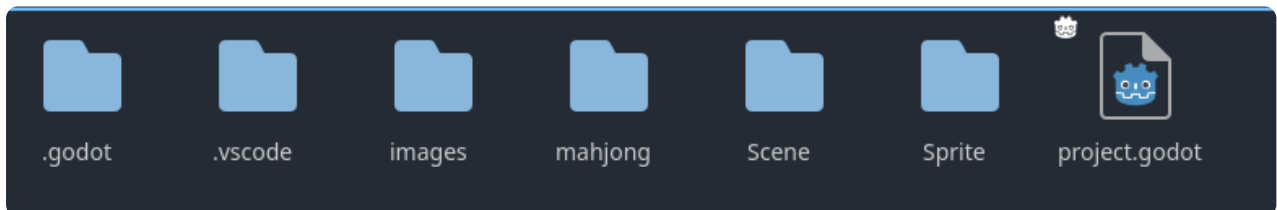
接下來CD中會有以上6個檔案，第1個是本專案的code，第二個是godot4.2.1的執行檔案。

點開第2個資料夾會有以下2個執行檔，點開第一個執行檔。

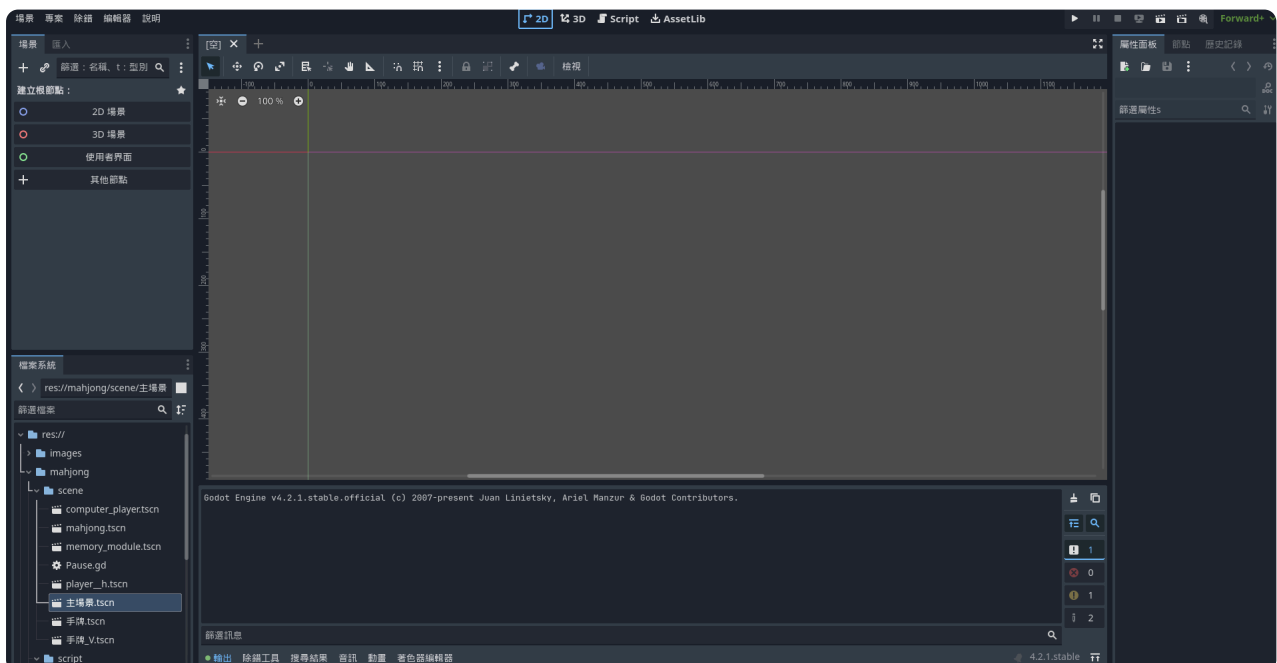
Godot_v4.2.1-stable_win64.exe	2023/12/12 下午 05:59	應用程式	105,664 KB
Godot_v4.2.1-stable_win64_console.exe	2023/12/12 下午 05:59	應用程式	192 KB

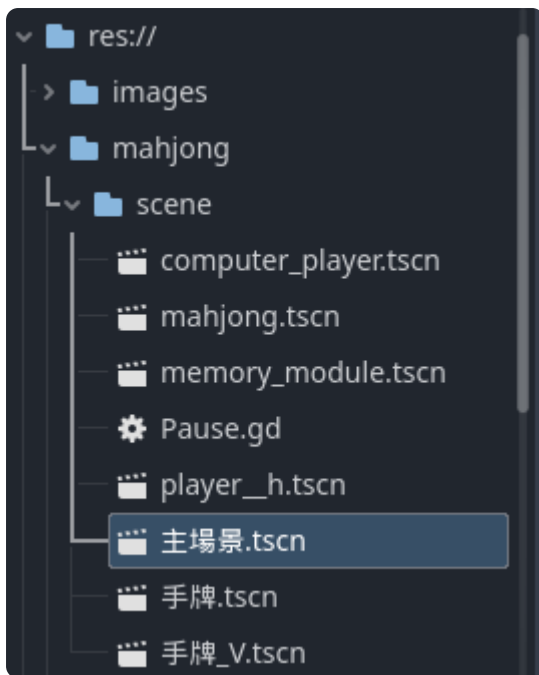


畫面上方會有匯入選項，將CD中的第一個資料夾點開會有7個檔案，將最右邊的匯入。

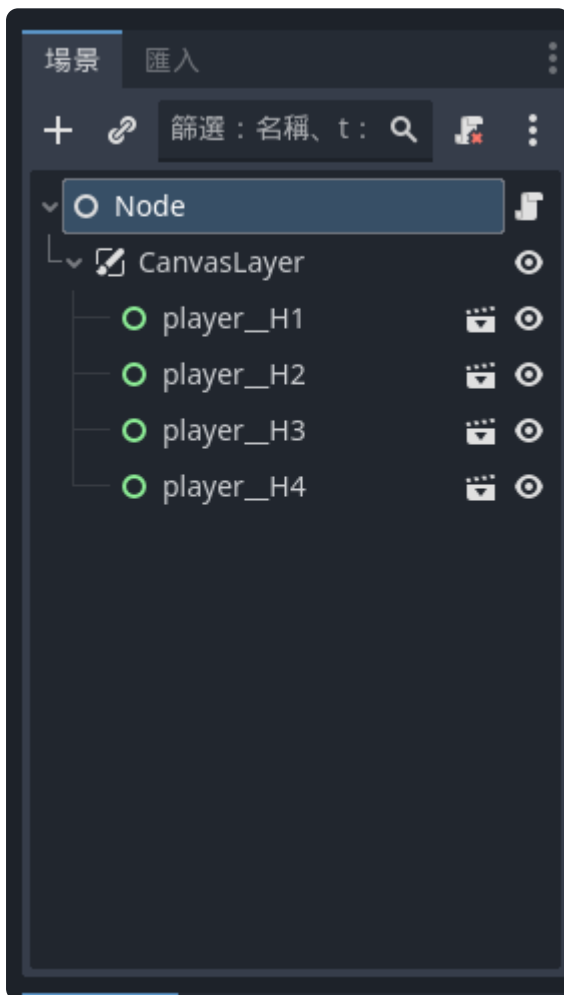


開啟後畫面如同下圖，優先將主場景雙擊點開。





資料夾上方會顯示如圖下，Node 右邊 script 點開來會能夠編輯程式碼。



```
17 | ...
18 | var apiKey = "請輸入openai的api key"
19 | ...
```

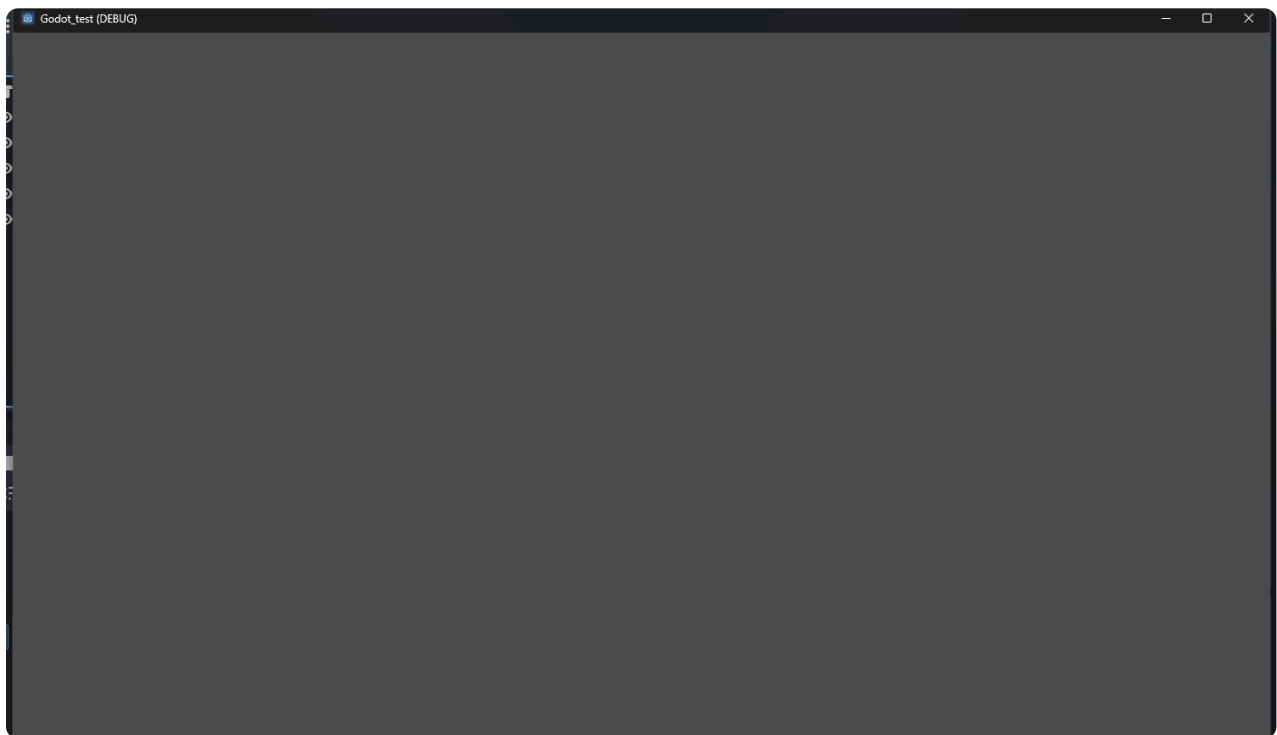
下方的code 82行 能夠更改目標，本程式碼是基於目標生成計畫在執行，所以目標對gpt的結果有密切的關係，83行 apiKey、model、生成的資料名稱、個性設定、目標。

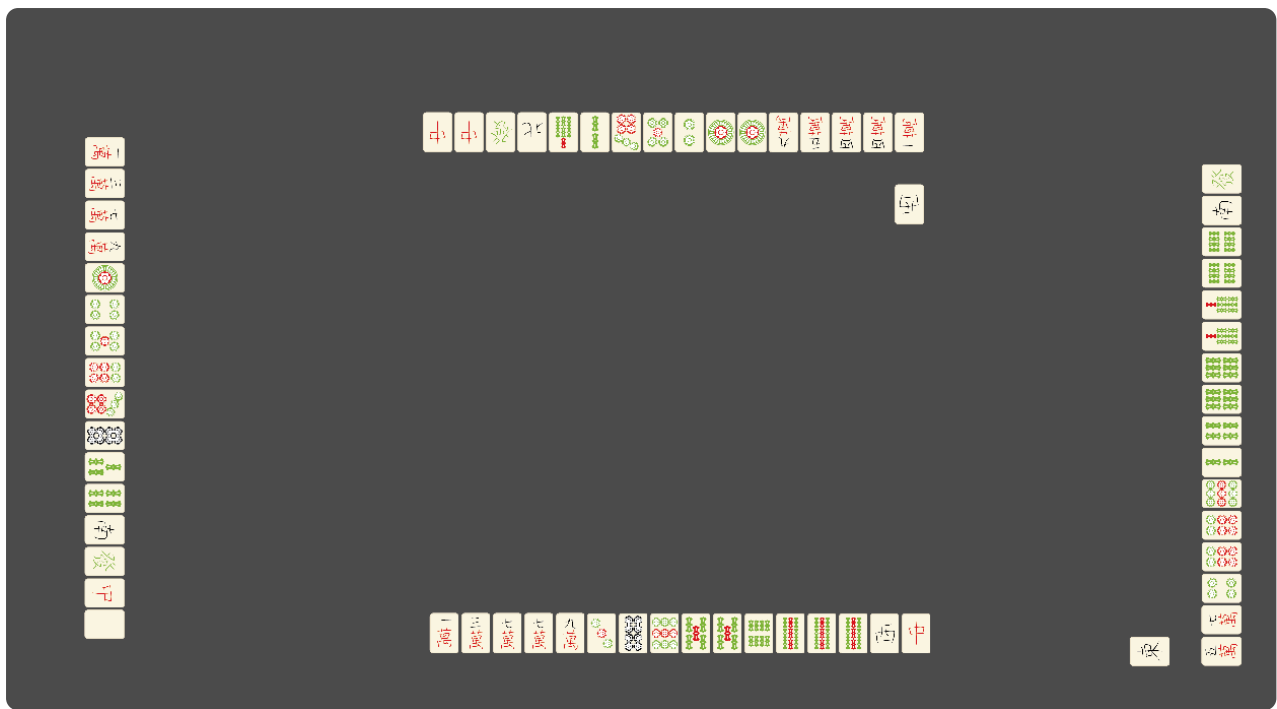
```
82 | var goal:Array[String] = ["最佳的出牌"]  
83 | players[i].gpt_inti(apiKey,"gpt-4o-mini","p1","你正在玩麻將你是一名沉著冷靜的
```

設定好後，畫面的右上角能夠執行當前的場景。



執行後如下圖一片空白是正常的，按下enter按鍵變會開始





接下來是如何觀測，如同下圖，可以分為2個部分，上部份便是將場景內容文字化，下方是gpt生成的計畫。

你的手牌：一萬,三萬,七萬,七萬,九萬,三筒,八筒,九筒,一條,五條,五條,八條,九條,九條,中,六筒
上家打出去的牌：南,發
對家打出去的牌：西,北
下家打出去的牌：東,南

根據目前的手牌和牌局情況，我們需要考慮如何最佳化組成順子、刻子及眼睛。以下是我的計畫思考：

- **分析手牌**：**
 - 萬子：一萬、三萬、七萬、七萬、九萬
 - 筒子：三筒、八筒、九筒、六筒
 - 條子：一條、五條、五條、八條、九條、九條、九條
 - 風牌：中
- **可能的組合**：**
 - ****順子**：**
 - 七萬、八萬、九萬（未完全，缺八萬）
 - 三筒、八筒、九筒（未完全，缺四筒）
 - ****刻子**：**
 - 五條、五條（可形成一個刻子，還有一張五條可作為丟棄）
 - 九條、九條、九條（已經形成了刻子）
 - ****眼睛**：**
 - 可以用兩張五條或兩張中作為眼睛。
- **出牌建議**：**
 - 現在的手牌中，最不容易組成有效的牌型的是「六筒」，因為它不與其他牌形成任何明顯的組合，且對於順子和刻子的組成幫助不大。

綜合以上分析，我的計畫是丟棄「六筒」，這樣可以保持手牌中其他牌的靈活性，增加胡牌的可能性。

這一局結束會在下方顯示gpt這回合打了些甚麼 胡牌 放槍 回合(從0開始算)

```
{ "draw_card": [[1, 0, 0, 1, 2, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 2, 0, 1, 2, 0, 1, 0, 0]], "win": 0, "lose": 0, "round": 0 }
```

接下來是其餘的設定調整。

```

27 var myseed = ["1"],["2"],["3"],["4"],["5"],["6"],["7"],["8"],["9"],["10"]]
.
.
.
66 if times == myseed.size() : return
67 var my_seed = myseed[times].hash()
68 seed(my_seed)
69 times += 1
# 如果打算全隨機請將 66 ~ 69 註解 將 71取消註解 便能夠seed隨機 · 27 是控制隨機的種子

71 #randomize()

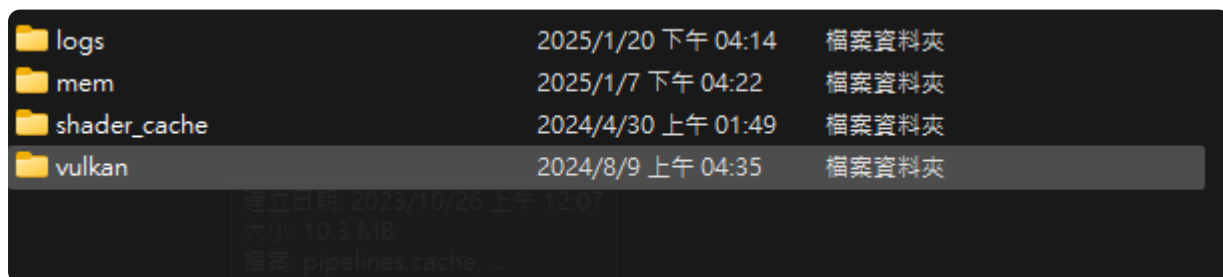
```

```

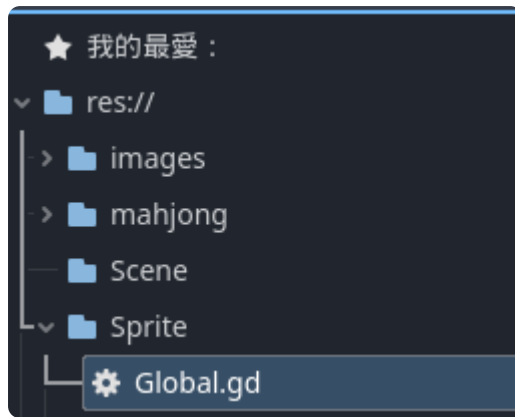
197 var model_name = "gpt-4o-無策略"
198 var Main_mem_path = "user://mem/" + "gpt/" + model_name + ".json"
199 #控制 輸出檔案內容如下
200 #{ "draw_card": [[1, 0, 0, 1, 2, 0, 1, 0, 1, 0, 0, 1, 1,
201 #0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 2, 0, 1, 2, 0, 1, 0, 0]],
202 #"win": 0, "lose": 0, "round": 0 }

```

點開此便能夠找到檔案位置。

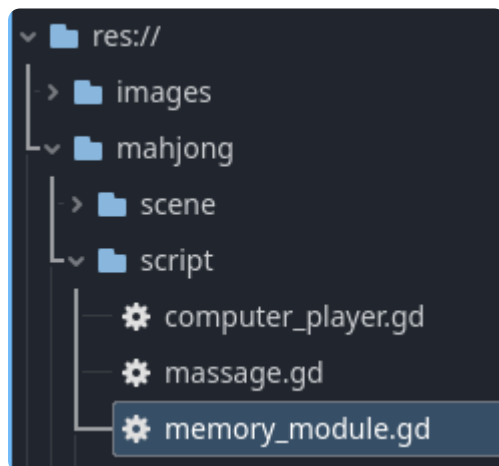


接著換設定對話內容的script。



```
94 var question : Dictionary = {
95   "Discard_a_tile" : ["請根據計畫的內容，打出一張手牌，精簡回答要打出的手牌。"],
96   # "Generate_plan" : ["請根據目標的內容，生成一個以目標記憶體為方向的計畫。ex.目標"],
97   "Generate_plan" : ["針對目標,生成一個自己的計畫，思考利用手牌組牌，輸出要從手牌丟棄"],
98   "Check_plan" : ["針對目標,生成一個自己的計畫思考利用手牌組牌，輸出要從手牌丟棄"],
99   "Generate_reflection" : ["請根據這把的內容，檢討這一把的問題在何處，整理成反思"],
100  "event_choice" : ["請根據選擇的內容，回答是否要做出選擇，你所有可以回答的選項有"],
101  "eat_choice" : ["請根據選擇的內容，回答是否要做出選擇，你所有可以回答的選項有[A"],
102  "error_response" : ["你回答的內容有誤，請重新回答。"], ["system", "main", "env"],
103  #左邊為問gpt時的問題，右邊是需要傳入的記憶模塊
104 }
```

system 模塊內容設定，



```
87 System_mem = {  
    "role": "system",  
    "content": "麻將的基本目標是以最快的方式將手中的牌組成特定的組合達到「胡牌」  
        1. **順子**：由三張連續的同花色(條、萬、筒)數字牌組成。例如：  
        .  
        .  
        .  
        .  
        .  
        '從你的手牌'選一張牌丟到棄牌區，請切記只能從'你的手牌'丟牌!!!"  
127 }
```

